A 2D Conditional Random Fields Model for Web Information Extraction*

Jun Zhu[1*], Zaiqing Nie[2], Ji-Rong Wen[2], Bo Zhang[1], Wei-Ying Ma[2]

[1]Tsinghua University, Beijing, China

[2]Microsoft Research Asia, Beijing, China

[1]{jun-zhu, dcszb}@mail.tsinghua.edu.cn, [2]{t-znie, jrwen, wyma}@microsoft.com

Abstract

The Web contains an abundance of useful semi-structured information about real world objects, and our empirical study shows that strong sequence characteristics exist for the Web information about the objects of the same type across different Web sites. This paper introduces a two dimensional Conditional Random Fields model, incorporating the sequence characteristics and the 2D neighborhood dependencies, to automatically extract object information from the Web. We also present the experimental results comparing our model with the linear-chain CRF model in the domain of product information extraction. The experimental results show that our model significantly outperforms existing CRF models.

1. Introduction

While the Web is traditionally used for hypertext publishing and accessing, there are actually various kinds of objects embedded in static Web pages and dynamic Web pages generated from online Web databases. There is a great opportunity for us to extract and integrate all the related Web information about the same object together as an information unit. These information units are called *Web objects* in (Nie et al., 2005). Typical Web objects are products, people, papers, organizations, etc. Commonly, objects of the same type obey the same structure or schema. We can imagine that once these objects are extracted and integrated from the Web, some large databases can be constructed to perform further knowledge discovery and data management tasks. This paper studies how to extend the existing information extraction techniques to automatically extract object information from Web pages.

The information about an object in a Web page is usually grouped together as a block, as shown in Figure 1. Using existing Web page segmentation technologies (Cai et al., 2004; Liu et al., 2003), we can automatically detect these Web object blocks, which are further segmented into atomic extraction entities called *object elements*. Each object element provides (partial) information about a single attribute of the Web object. Given an object block B with a set of elements $B = \{e_i\}_{i=1}^{T}$, the Web object extraction task is to assign an attribute name from the attribute set $A = \{a_i\}_{i=1}^{m}$ to each object element e_i.

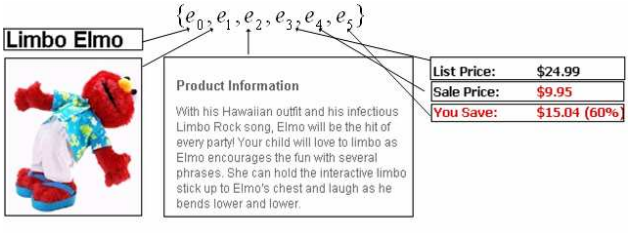

Figure 1. An object block with 6 elements in a Web page

Intuitively, the sequence order of the attribute values appearing in an object block is similar to that of another block about an object of the same type. If this sequence characteristic is suitable for Web objects of interests, Condition Random Fields (CRF) models (Lafferty et al., 2001) are the state of the art approaches in information extraction taking the advantage of the sequence characteristics to do better labeling. To show that the sequence order is similar among the Web objects of the same type, we have conducted some statistical studies over a set of randomly selected Web sites. The results (see table 1) show that strong sequence characteristics exist for Web objects of the same type across different Web sites.

However, in order to use a linear-chain structured CRF model for Web object information, we have to first convert a two dimensional object block into a sequence of object elements. Given the two dimensional nature of the object blocks, how to sequentialize them in a meaningful way can be very challenging. Moreover, as shown by our empirical evaluations, using the 2D neighborhood dependencies (*i.e.* interactions between labels of the neighbors of the current element in both vertical and horizontal directions) in Web object extraction could significantly improve the extraction accuracy, and only considering one dimensional neighbor will lead to ineffective solutions.

In this paper, we first propose a two dimensional Conditional Random Fields model with the graph as a 2D lattice (as see Figure 2). Then we deduce the forward-backward algorithm for 2D CRFs based on the matrices expressed conditional distribution for efficient parameter estimation and labeling. Since the model is two dimensional, exact inference can be very expensive, so a

* Technical Report. Microsoft Research. **MSR-TR-2005-44**

* This work is done when the author is visiting Microsoft Research Asia.

suboptimal method is used to perform approximate inference. To handle the irregular neighborhood dependencies caused by the elements' arbitrary sizes in a Web page, we introduce the concepts of *virtual elements* and *empty elements* to map an object block with arbitrary sized elements into a 2D lattice.

We compare our model with linear-chain CRF models for product information extraction and the experimental results show that our model significantly outperforms linear-chain CRF models in scenarios with 2D neighborhood dependencies.

The rest of this paper is organized as follows. We discuss the related work in the next section. In section 3, a 2D CRF model is presented, the parameter estimation and labeling methods are discussed. Section 4 presents web object extraction in a two dimensional sense. In section 5, we setup our experiments in the domain of product information extraction and give the experimental results. Section 6 brings this paper to a conclusion.

2. Related Work

For information extraction, there have been many probabilistic models. In the past, as the IE was mainly taken as a sequential labeling task, so the models used were generally linear-chain models for simplicities. Among this type of models are HMMs, MEMMs, CRFs, etc. In the following, we use x and y to denote the observation and label sequences respectively. HMMs (Bikel et al., 1997; Leek, 2000) or Hidden Markov Models, are generative models which define the joint probability distribution $p(\mathrm{x}, \mathrm{y})$. But for information extraction, the conditional distribution $p(\mathrm{y} \mid \mathrm{x})$ is of interest, so these models must enumerate all possible observation sequences to compute the conditional distribution. If the observations have long-distance dependencies, this is intractable. In order to achieve computational tractability, the independence assumption is made: the observation at time t is conditionally independent from other observations given the state at time t . This assumption is too strong when the observations have long-distance dependencies or multiple interacting features. To relax the strong assumption, conditional models are proposed. A conditional model specifies the probabilities of possible label sequences given an observation sequence. Therefore, it does not take any effort on modeling the observations and the conditional probability over the label sequences can depend on arbitrary, over-lapping features of the observations. MEMMs (McCallum, Freitag & Pereira, 2000) or maximum entropy markov models are models of this type. MEMMs use the state-observation transition distribution $p(\mathrm{y}_t \mid \mathrm{y}_{t-1}, \mathrm{x}_t)$ to replace the state transition probability $p(\mathrm{y}_t \mid \mathrm{y}_{t-1})$ and the observation probability $p(\mathrm{x}_t \mid \mathrm{y}_t)$ in HMMs. However, as pointed out by Lafferty, McCallum and Pereira (2001), MEMMs and some discriminative models are brittle to the Label Bias Problem. For the per-state normalization of the next state distribution, MEMMs take little notice of observations at the states which have low-entropy next state distribution and just ignore the observations at the states which have only one next state.

CRFs or conditional random fields were introduced by Lafferty et al. (2001) to take the advantages of conditional models and also to avoid the Label Bias problem suffered by MEMMs. CRFs are undirected graphical models, so the single joint probability distribution over the label sequence given the observation rather than the per-state distributions over the next states given the current state can be specified. CRF models have been shown to outperform other models in modeling sequential data (Sha & Pereira, 2003; Peng & McCallum, 2004). Dynamic conditional random fields (DCRFs) (Sutton et al., 2004) are a generalization of linear-chain CRFs to represent complex interaction between labels in sequence labeling. As a specific model, the factorial CRF model is used for a natural-language chunking task and the approximate inference is performed using loopy belief propagation (Murphy, Weiss & Jordan, 2002).

Previous work on 2D models was mainly carried out in the domains of Image processing and Computer Vision. Among these models are 2D HMMs (Li, Najmi & Gray, 2000), MRF models (Besag, 1974; Li, 2001), and DRFs (Kumar & Hebert, 2003; Kumar & Hebert, 2004). The 2D Hidden Markov models proposed by Li et al. (2000) for image classification are also generative models, so some independence assumptions are made for computational tractability. The fast algorithms developed to efficiently estimate the model and perform classification are worth noting. Markov Random Fields are generally used in a probabilistic generative framework, but unlike 2D HMMs, MRFs model the prior distribution $p(\mathrm{y})$ over labels as a markov random field. For computational tractability, the likelihood model $p(\mathrm{x} \mid \mathrm{y})$ is usually also assumed to have a factorized form as in the 2D HMMs. Some researchers (Cheng & Bouman, 2001; Wilson & Li, 2003) have noticed that this assumption is too restrictive for some applications. To take advantages of the conditional models, DRFs or Discriminative Random Fields were proposed by Kumar et al. (2003) in the case of binary image classification, which are based on CRFs and model the association potential as local discriminative model as well as the interactions between the neighboring sites on 2D lattices. To make the parameter learning a convex problem, a simplified interaction potential form was proposed by Kumar et al. (2004). It has been shown that DRFs outperform MRFs in the natural image classification. However, the proposed DRFs are concerned with binary classification, so they can't be applied to our application.

3. 2D Conditional Random Fields

In this section we first introduce the basic concepts of the

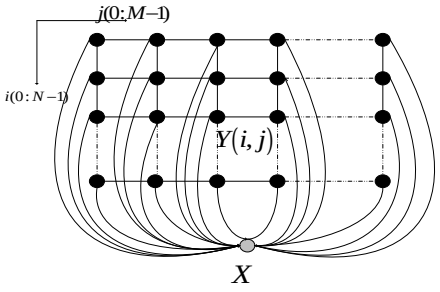

Figure 2. The graphical structure of 2D CRF, where X is the observed random variable; $Y(i, j)$ is the state variable which is indexed by the vertex (i, j) in the 2D lattice.

linear–chain Conditional Random Fields models, and then we discuss the conditional probability over labels of a 2D CRF model, and finally we discuss how to estimate the parameters and how to perform labeling.

3.1 Linear-chain CRFs

Conditional random fields (Lafferty et al., 2001) are undirected graphs. As defined before, $X = \{X_i\}_{i \in V}$ is a random variable over the observed data to be labeled, and $Y = \{Y_i\}_{i \in V}$ is the corresponding random variable over labels, where Y_i range over a finite label alphabet Y . The random variables X and Y are jointly distributed, but in a discriminative framework, a conditional model $p(Y \mid X)$ is constructed from paired observations and labels, and the marginal $p(X)$ is not explicitly modeled.

CRF Definition:

Let $G = (V, E)$ be an undirected graph such that Y is indexed by the vertices of G. Then (X, Y) is said to be a conditional random field if, when conditioned on X, the random variables Y_i obey the Markov property with respect to the graph: $p(Y_i \mid X, Y_{V-\{i\}}) = p(Y_i \mid X, Y_{N_i})$ where $V - \{i\}$ is the set of nodes in the graph except the node i, N_i is the set of neighbors of the node i in G.

Thus, a CRF is a random field globally conditioned on the observations X . By the Hammersley-Clifford Theorem (Hammersley & Clifford, 1971), the joint distribution over the labels y given the observations x of a linear-chain CRF has the form,

$$p(\mathrm{y} \mid \mathrm{x}) = \frac{1}{Z(\mathrm{x})} \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$

where $\mathrm{y}|_S$ is the set of components of y associated with the vertices in the sub-graph S , f_k is a transition feature function and g_k is a state feature function, λ_k and μ_k are parameters to be estimated from the training data, $Z(\mathrm{x})$ is the normalization factor, also known as partition function, which has the form,

$$Z(\mathrm{x}) = \sum_{\mathrm{y}} \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$

3.2 2D CRFs

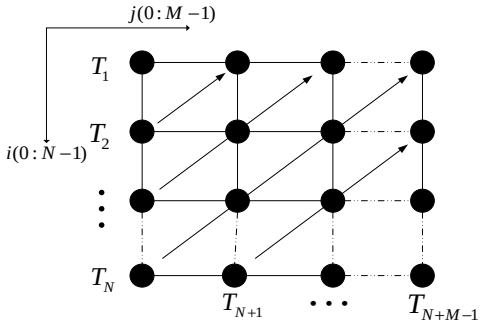

Figure 3. The diagonal state sequences of a 2D lattice

For 2D CRFs, the underlying graph is a 2D lattice (see Figure 2). The cliques in this graph are edges and vertices. Therefore, the conditional distribution has the same form as the linear-chain CRFs, but with $E = E_v \cup E_h$, and E_v , E_h are the sets of vertically and horizontally oriented edges respectively.

For linear-chain CRF (Lafferty et al., 2001), the forward-backward algorithm is used to carry out efficient inference based on the matrix formatted conditional distribution. Similarly, for 2D CRF, we present the conditional distribution in a matrix form to deduce the forward-backward vectors for efficient parameter estimation and labeling (see Section 3.3 & 3.4).

We first introduce some notations we use as follows:

1) The sequence of states on the diagonal i , $\{Y_{i,0}, Y_{i-1,1}, \cdots, Y_{0,i}\}$ is denoted by T_{i+1} , as shown in Figure 3.
2) The label sequence on diagonal i , $\{\mathrm{y}_{i,0}, \mathrm{y}_{i-1,1}, \cdots, \mathrm{y}_{0,i}\}$ is denoted by y_{i+1} .
3) Two special state sequences are added: $T_0 = start$ and $T_{N+M} = stop$.
4) The diagonal on which the state $Y_{i,j}$ lies is denoted by $\Delta(i, j)$.
5) The set of indices of the states on diagonal d , $\{(i, j), \Delta(i, j) = d\}$ is denoted by $I(d)$.
6) The set of edges between diagonals $d-1$ and d $\{((i', j'), (i, j)) \in E : (i', j') \in I(d-1) \text{ and } (i, j) \in I(d)\}$ is denoted by $E(d)$.

For each diagonal position i , we define the $|Y|^{S_{i-1}} \times |Y|^{S_i}$ matrix random variable $M_i(\mathrm{x}) = [M_i(\mathrm{y}'_{i-1}, \mathrm{y}_i \mid \mathrm{x})]$ by

$$M_i(\mathrm{y}'_{i-1}, \mathrm{y}_i \mid \mathrm{x}) = \exp(\Lambda_i(\mathrm{y}'_{i-1}, \mathrm{y}_i \mid \mathrm{x}))$$

$$\Lambda_i(\mathrm{y}'_{i-1}, \mathrm{y}_i \mid \mathrm{x}) = \sum_{e \in E(i),k} \lambda_k f_k(e, \mathrm{y}'_{w,u}, \mathrm{y}_{w,u}, \mathrm{x}) + \sum_{v \in I(i),k} \mu_k g_k(v, \mathrm{y}_{w,u}, \mathrm{x}) \quad \cdots (1)$$

where S_{i-1} and S_i are the state numbers on diagonals $i-1$ and i respectively, and $e = ((w', u'), (w, u))$ is an edge between diagonal $i-1$ and diagonal i , and $v = (w, u)$ is a vertex on diagonal i .

Thus, when given the observations x and the parameters, the matrices can be computed as needed. Then the normalization factor $Z(\mathrm{x})$ can be expressed as the $(start, stop)$ entry of the product of these matrices:

$$Z(\mathrm{x}) = \left(M_1(\mathrm{x})M_2(\mathrm{x})\cdots M_{M+N}(\mathrm{x})\right)_{(start,stop)}$$

So, the conditional probability over labels y given the observations x has the form,

$$p(\mathrm{y} \mid \mathrm{x}) = \frac{\prod_{i=1}^{M+N} M_i(\mathrm{y}_{i-1}, \mathrm{y}_i \mid \mathrm{x})}{\left(\prod_{i=1}^{M+N} M_i(\mathrm{x})\right)_{(start,stop)}} \qquad \cdots (2)$$

where $\mathrm{y}_0 = start$ and $\mathrm{y}_{N+M} = stop$.

3.3 Parameter Estimation

Given the training data $D = \{(\mathrm{y}^i, \mathrm{x}^i)\}_{i=1}^N$ with the empirical distribution $\tilde{p}(\mathrm{x}, \mathrm{y})$, the log-likelihood of $\tilde{p}(\mathrm{x}, \mathrm{y})$ with respect to a conditional model $p(\mathrm{y} \mid \mathrm{x}, \Theta)$ is defined as,

$$L(\Theta) = \prod_{\mathrm{x},\mathrm{y}} \tilde{p}(\mathrm{x}, \mathrm{y}) \log p(\mathrm{y} \mid \mathrm{x}, \Theta)$$

The parameter estimation problem is to find a set of parameters $\{\lambda_1, \lambda_2, \cdots; \mu_1, \mu_2, \cdots\}$ that optimize the convex log-likelihood function. The function can be optimized by the techniques used in other maximum-entropy models (Lafferty et al., 2001; Berger et al., 1996). We choose L-BFGS (Liu & Nocedal, 1989) for that it has been shown to outperform other optimization algorithms for linear-chain CRFs (Malouf, 2002; Sha et al., 2003). Each element of the gradient vector is given by

$$\frac{\partial L(\Theta)}{\partial \lambda_k} = E_{\tilde{p}(\mathrm{x},\mathrm{y})}[f_k] - E_{p(\mathrm{y}|\mathrm{x},\Theta)}[f_k]$$

where $E_{\tilde{p}(\mathrm{x},\mathrm{y})}[f_k]$ is the expectation with respect to the empirical distribution, and $E_{p(\mathrm{y}|\mathrm{x},\Theta)}[f_k]$ is the expectation with respect to the conditional model distribution. For transition feature functions,

$$E_{p(\mathrm{y}|\mathrm{x},\Theta)}[f_k] = \sum_{\mathrm{x}} \tilde{p}(\mathrm{x}) \sum_{e \in E} \sum_{y'_{i',j'}, y_{i,j}} p(y'_{i',j'}, y_{i,j} \mid \mathrm{x}) f_k(e, y'_{i',j'}, y_{i,j}, \mathrm{x})$$

where $e = ((i', j'), (i, j))$ is an edge, and for state feature functions,

$$E_{p(\mathrm{y}|\mathrm{x},\Theta)}[g_k] = \sum_{\mathrm{x}} \tilde{p}(\mathrm{x}) \sum_{v=(i,j) \in V} \sum_{y_{i,j}} p(y_{i,j} \mid \mathrm{x}) g_k(v, y_{i,j}, \mathrm{x})$$

So the computation of the marginal probabilities, which are needed to compute the gradients at every iteration, is the main contribution to the complexity. The idea of forward-backward algorithm can be extended here to simplify the computation. As the conditional distribution has the form in equation (2), the state sequence T_i is in fact an "isolating" element in the expansion of $p(Y \mid X)$, which plays the same role of a state at a single unit of time in linear-chain CRFs.

For each diagonal index $d = 0, \cdots, M+N$, the forward vectors $\alpha_d(\mathrm{x})$ are defined with base case

$$\alpha_0(\mathrm{y}_0 \mid \mathrm{x}) = \begin{cases} 1 & if \quad \mathrm{y}_0 = start \\ 0 & otherwise \end{cases}$$

and with recurrence $\alpha_d(\mathrm{x}) = \alpha_{d-1}(\mathrm{x}) M_d(\mathrm{x})$

Similarly, the backward vectors $\beta_d(\mathrm{x})$ are defined by

$$\beta_{N+M}(\mathrm{y}_{M+N} \mid \mathrm{x}) = \begin{cases} 1 & if \ \mathrm{y}_{M+N} = stop \\ 0 & otherwise \end{cases}$$

and $$\beta_d(\mathrm{x})^{\mathrm{T}} = M_{d+1}(\mathrm{x}) \beta_{d+1}(\mathrm{x})$$

Thus, the marginal probability of being in label sequence y_d at diagonal d given the observations x is

$$p(\mathrm{y}_d \mid \mathrm{x}) = \frac{\alpha_d(\mathrm{y}_d \mid \mathrm{x}) \beta_d(\mathrm{y}_d \mid \mathrm{x})}{Z(\mathrm{x})}$$

So the marginal probability of being in state $y_{i,j}$ at $T_{i,j}$ on diagonal d is

$$p(y_{i,j} \mid \mathrm{x}) = \sum_{\mathrm{y}_d : \mathrm{y}_d(i,j) = y_{i,j}} p(\mathrm{y}_d \mid \mathrm{x})$$

Similarly, the marginal probability of being in label sequence y'_{d-1} at diagonal $d-1$ and y_d at diagonal d given the observations x is

$$p(\mathrm{y}'_{d-1}, \mathrm{y}_d \mid \mathrm{x}) = \frac{\alpha_{d-1}(\mathrm{y}'_{d-1} \mid \mathrm{x}) M_d(\mathrm{y}'_{d-1}, \mathrm{y}_d \mid \mathrm{x}) \beta_d(\mathrm{y}_d \mid \mathrm{x})}{Z(\mathrm{x})}$$

So the marginal probability of being in state $y'_{i',j'}$ at $T_{i',j'}$ and $y_{i,j}$ at $T_{i,j}$ is,

$$p(y'_{i',j'}, y_{i,j} \mid \mathrm{x}) = \sum_{\mathrm{y}'_{d-1} : \mathrm{y}'_{d-1}(i',j') = y'_{i',j'}} \sum_{\mathrm{y}_d : \mathrm{y}_d(i,j) = y_{i,j}} p(\mathrm{y}'_{d-1}, \mathrm{y}_d \mid \mathrm{x})$$

where $((i', j'), (i, j))$ is an edge between diagonals $d-1$ and d.

3.4 Labeling

Labeling is the task to find a labels y^* that best describes the observations x, that is,

$$\mathrm{y}^* = \max_{\mathrm{y}} p(\mathrm{y} \mid \mathrm{x})$$

Dynamic programming algorithm is the desirable method for this problem. A variant of viterbi algorithm is used by Lafferty et al. (2001) for linear-chain CRFs. In the 2D case, viterbi algorithm can also be extended for the labeling task using the "isolating" element T_i (Li et al., 2000). The difference from the normal Viterbi algorithm is that the number of possible state sequences at every position in the viterbi transition diagram is exponential to the number of states on T_i.

4. Modeling an Object Block

As described in the introduction section, an object block is composed of some atomic extracted entries called object elements. The object blocks and their elements can be located using some existing web page segmentation technologies like (Cai et al., 2004; Liu et al., 2003). However, when using the proposed 2D CRF model to model the interactions between neighborhood states for Web object extraction, we need to handle the irregular neighborhood dependencies caused by the elements' arbitrary sizes in a Web page. Take the block in Figure 1 as an example, the element e_2 is so large that the elements e_1, e_3, e_4 and e_5 are its neighbors. But for the lattice structured model, if we associate each element with only one state as in Figure 4(a), the association result can not represent the desired neighborhood dependencies between state pairs (e_2, e_4) and (e_2, e_5). Moreover, we should not further segment them into smaller elements to represent the neighborhood dependencies because they are atomic extracted entries. Thus, we use *virtual elements* and *empty elements* to handle this problem. By denoting the object elements as *real elements*, the *virtual elements* are defined as the mirrors of the real elements which must have the same attributes. The *empty elements* are introduced just for the denotation of missing states, which are ignored during the inference. In the following, we denote the *empty element* by e_{-1}, and the *virtual element* of real element e_i by e_i^*, where i is a unique index ranging from 0 to $n-1$ and n is the number of elements in a block. To

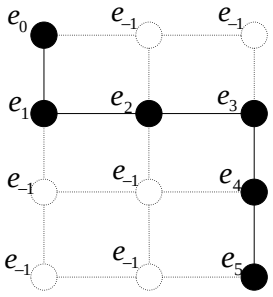

Figure 4(a). Direct association result

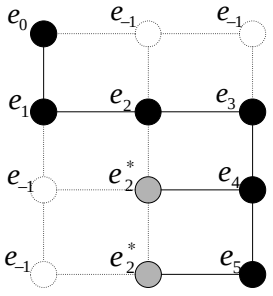

Figure 4(b). Association result with virtual elements

model the neighborhood dependencies in the 2D lattice, we define four neighbors (left, top, right and bottom) for each element as the neighbors of the state with which it is associated. The indices of the neighbors are denoted by a quad-tuple (l_i, t_i, r_i, b_i). If e_i has only one left, top, right, or bottom neighbor, then the corresponding index is just the index of that neighbor; If e_i has more than one left or right neighbors, then l_i or r_i is the index of the highest one; If e_i has more than one top or bottom neighbors, then t_i or b_i is the index of the most left one. If a neighbor doesn't exist, the index is –1 accordingly. Thus, the neighbors of each element in Figure 1 are:

$$
\begin{aligned}
e_0 &: (-1,-1,-1,1)\\
e_1 &: (-1,0,2,-1)\\
e_2 &: (1,-1,3,-1)\\
e_3 &: (2,-1,-1,4)\\
e_4 &: (2,3,-1,5)\\
e_5 &: (2,4,-1,-1)
\end{aligned}
$$

The association result with virtual elements is shown in Figure 4(b). The states are *real*, *virtual* or *empty states* if the elements associated with them are *real*, *virtual* or *empty* elements. Since the empty states are ignored during inference, a diagonal state sequence is composed of the *real* and *virtual* states on that diagonal. Thus, the diagonal state sequences in Figure 4(b) are:

$$
\begin{aligned}
T_0 &: 0(0,0)\\
T_1 &: 1(1,0)\\
T_2 &: 2(1,1)\\
T_3 &: 2(2,1)^*, 3(1,2)\\
T_4 &: 2(3,1)^*, 4(2,2)\\
T_5 &: 5(3,2)
\end{aligned}
$$

The virtual states are half-real for that an edge associated with them is normal clique if its another end is associated with a different real element; otherwise it's a virtual clique. In Figure 4(b), we have denoted *normal cliques* as solid edges and *virtual cliques* as dotted edges. The *virtual cliques* do not contribute to the probability distribution, but indicate strong constraints between the same virtual states or between the virtual states and their real states that they must have the same state values when a transition takes place. Thus, equation (1) can be reformed as:

$$
\Lambda_i\left(y'_{i-1}, y_i \mid x\right) = \begin{cases} -\infty, & \exists ((i',j'),(i,j)) \in E_v(i) \;\; s.t. \; y_{i',j'} \neq y_{i,j} \\ \sum_{e \in E_r(i),k} \lambda_k f_k(e, y'_{w',u'}, y_{w,u}, x) + \sum_{v \in I_r(i),k} \mu_k g_k(v, y_{w,u}, x), & \text{otherwise} \end{cases}
$$

where $E_v(i)$ is the set of virtual cliques between diagonals $i-1$ and i, $E_r(i)$ is the set of normal cliques, and $I_r(i)$ is the set of indices of the real states on diagonal i and $e = ((w', u'), (w, u))$ is an edge between

diagonals $i-1$ and i, and $v=(w,u)$ is a vertex on diagonal i. So, the transition matrix from T_2 to T_3 with $Y=\{0,1\}$ is,

$$M_3(\mathrm{x}) = T_2 : \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccc} T_3: & 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} m_{00} & m_{01} & 0 & 0 \\ 0 & 0 & m_{12} & m_{13} \end{bmatrix}_{2\times 4} \end{array}$$

The elements m_{02}, m_{03}, m_{10} and m_{11} are zeros for the strong constraints between states $T_{1,1}$ and $T_{2,1}$.

As the dimensions of the transition matrices $M_i(\mathrm{x})$ are exponential to the state numbers on diagonals $i-1$ and i respectively, the computational complexity is really high. In order to achieve polynomial-time complexity, we use the path-constrained suboptimal method proposed by Li et al. (2000). We apply this method to compute the approximate gradients in L-BFGS algorithm to train our model and to find the best state sequence using variable-state Viterbi algorithm.

5. Experimental Studies

In this section, we first conducted some statistical work to demonstrate that strong sequence characteristics exist for Web objects of the same type across different Web sites. Then we compare our model with linear-chain CRF in the domain of product information extraction. The experimental results show that our model outperforms sequential models in Web object extraction.

5.1 Statistical Results

To show that the sequence characteristics are universal in Web object blocks, we conducted some statistical work in two types of Web objects: products and researchers' homepages. We randomly selected 100 product pages containing 964 product blocks from different Web sites and 120 homepages. Some key attributes are surveyed for each type. For product objects, the attributes "name", "image", "price" and "description" are considered and for researchers' homepages, we considered the attributes "name", "telephone", "email" and "address". We decide the sequence order of the elements in a web page in a top-down and left-right manner based on their position information. Basically, the element in the top level will be ahead of the all the elements below it and for the elements at the same level, the left elements will be ahead of their right elements. In Table 1 we show the statistics about the sequence orders of the attribute pairs for objects from both product pages and homepages.

The statistical results show that strong sequence characteristics universally exist among most attribute pairs in both types of objects. For example, a product's name is always ahead of its description in all the pages.

Table 1. Statistical results from 100 randomly selected product web pages and total 964 product blocks. the results are from 120 randomly selected homepages. We have used "DESC" instead of "DESCRIPTION" and "TEL" instead of "TELEPHONE" for space.

PRODUCT	BEFORE	HOMEPAGE	BEFORE
(NAME, DESC)	1.000	(NAME, TEL)	1.000
(NAME, PRICE)	0.987	(NAME, EMAIL)	1.000
(IMAGE, NAME)	0.941	(NAME, ADDRESS)	1.000
(IMAGE, PRICE)	0.964	(ADDRESS, EMAIL)	0.847
(IMAGE, DESC)	0.977	(ADDRESS, TEL)	0.906

5.2 Performance Evaluation

To fully test our model, we carried out our experiments in the domain of product object information extraction for its plentiful spatial information. In the experiments, we considered four attributes, "name", "image", "price", and "description".

5.2.1 DATASETS

We setup our datasets with 572 randomly crawled product Web pages [1], and we use the Web page segmentation technology (Cai et al., 2004) to segment the crawled Web pages and collect all the blocks that contain product information. The block elements can be further segmented using this technology at a finer granularity. An appropriate segmentation granularity is important when segmenting the elements, because either over-segmentation or less-segmentation will affect the extraction accuracy. In our experiments, we prefer over-segmentation, that is, we always prefer smaller elements when there is segmentation uncertainty.

We collect totally 2500 product blocks from 572 Web pages to form our data set. There are two types of these blocks, the first type is that whose elements are just in a sequence, do not have two dimensional dependencies in our model. So, for this type of data, our model really performs in a one dimensional sense as a linear-chain CRFs model. This data set is denoted by **ODS**. The second type is that whose elements have two dimensional dependencies when being associated with our model's states. This type of dataset is denoted by **TDS**. **ODS** contain 1000 blocks and **TDS** also contains 1000 blocks. The remaining 500 blocks form our training data. All the blocks are manually labeled.

5.2.2 MODEL CONSTRUCTION AND TRAINING

When testing our model, we focused on the effectiveness of using more spatial information for web information extraction. To compare our model with other sequential models, we chose linear-chain CRF model for its outstanding performance over other sequential models.

[1] URL of the data is omitted for double-blind reviewing.

We construct both the linear-chain CRF and 2D CRF models with the same set of feature functions. In our experiments, we used an html-parser to accurately get all the needed features, such as,

Text: the most commonly used information. For example, "price" must contain numbers and may also contain "$" or "%", etc.

Position: the most important information in our model to locate the object elements. The accurate position information is crucial for the association procedure.

Element's area: as necessary as the position information to find the neighborhood dependencies between elements. Currently we approximate the elements' areas by a quadrangle's area. The quadrangle's height and width can be got with the same parser.

And some other features, such as font, link-URL, image's source URL, etc, are also used to improve the accuracy.

Both models are trained on the same training data set. We use L-BFGS (Liu et al., 1989) algorithm to train our models. For linear-chain CRF model, the gradients are exactly computed, and for 2D CRF model, the gradients are approximately computed using the suboptimal method (Li et al., 2000). We used the convergence criterion,

$$\frac{\left|L\left(\Theta^{(k)}\right) - L\left(\Theta^{(k-1)}\right)\right|}{L\left(\Theta^{(k)}\right)} < \varepsilon$$

where, the relative tolerance is set $\varepsilon = 10^{-7}$ as in Malouf (2002). With 500 training samples and the same initial parameters, the linear-chain CRF model converges within 12 iterations, and the 2D CRF model converges within 13 iterations.

5.2.3 Experimental results

We compare our model with linear-chain CRF model on the data sets **ODS** and **TDS**. The performance is evaluated by *precision*, the percentage of returned elements that are correct; *recall*, the percentage of correct elements that are returned; and their harmonic mean *F1*, of each attribute. We also define two comprehensive evaluation criteria: (1) block instance accuracy as the percentage of blocks of which the key attributes (name, image, and price) are correctly labeled, (2) AVG_F1 as the average of *F1* values of different attributes.

5.2.4 Analysis

The experimental results (both table 2 and table 3) show that for the dataset **ODS**, there is no significant difference between linear-chain CRF model and 2D CRF model because of data's sequential properties. But for really two dimensional dataset **TDS**, the accuracy improvements of different attributes are significant. The precision and recall of attribute "name" are both improved by 12%. Although the precision of attribute "description" is not significantly improved, only by 2.1%, the recall is improved by 13.4%. For attributes "image" and "price", the improvements are not so significant, because these two attributes have notable state-information, that is, they can be well labeled only with the state feature functions. For example, prices must contain formatted numbers in the Web pages, and images are all with empty texts and non-empty image source URLs. Only using this information can give good results, so the neighborhood dependencies, which are represented by the transition feature functions in the model, does not contribute so much. From the average *F1*, we can also see the contribution of the spatial dependencies to improve the extraction accuracy. For one dimensional dataset **ODS**, the improvement of AVG_F1 is neglectable; but for **TDS**, the improvement is 7.7%. The block instance accuracy in table 3 says the same thing: the improvement of using 2D CRF on the dataset **TDS** is 15.6%, but the improvement on the dataset **ODS** is just 2.7%. Thus, the proposed model significantly outperforms linear-chain CRF models for the two dimensional Web information extraction.

Table 2. Precision, recall, F1 and AVG_F1 values of the attributes "NAME", "IMAGE", "PRICE" and "DESCRIPTION" on the datasets **TDS** and **ODS**. We have used "DESC" in stead of "DESCRIPTION" for space. 2D stands for 2D CRF model and Linear stands for linear-chain CRF model.

		TDS		**ODS**	
		2D	**Linear**	**2D**	**Linear**
Precision	NAME	**0.911**	0.790	**0.794**	0.762
	IMAGE	**0.963**	0.917	**0.993**	0.979
	PRICE	**0.969**	0.932	**0.977**	0.952
	DESC	**0.849**	0.828	**0.772**	0.813
Recall	NAME	**0.883**	0.762	**0.767**	0.734
	IMAGE	**0.963**	0.917	**0.993**	0.979
	PRICE	**0.919**	0.895	**0.942**	0.945
	DESC	**0.803**	0.669	**0.792**	0.816
F1	NAME	**0.897**	0.776	**0.781**	0.745
	IMAGE	**0.963**	0.917	**0.993**	0.979
	PRICE	**0.944**	0.913	**0.959**	0.949
	DESC	**0.824**	0.740	**0.782**	0.814
Avg_f1		**0.907**	0.837	**0.879**	0.872

Table 3. Block instance accuracy of linear-chain CRF and 2D CRF on datasets **ODS** and **TDS**.

	TDS	**ODS**
linear-chain CRF	0.600	0.755
2D CRF	0.756	0.782

6. Conclusions

In this paper, we propose a two dimensional Conditional Random Fields model. This model provides a way to incorporate 2D neighborhood dependencies to improve the performance for web information extraction. As the model is two dimensional, the exact inference can be very expensive, so a suboptimal method is used to efficiently estimate the model's parameters and to perform labeling task. When the model is applied for product information extraction, the experimental results show that our model significantly outperforms linear-chain CRF models.

References

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B,* 36:192-236.

Berger, A. L., Pietra, S. A. D., & Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22, 39-71.

Bikel, D., Miller, S., Schwartz, R., & Weischedel, R. (1997). Nymble: A high-performqance learning name-finder. In *Proc. Conf. on Applied Natural Language Processing*.

Cai, D., Yu, S., Wen, J. R., & Ma, W. Y. (2004). Block-based web search. *In ACM SIGIR Conference*, 2004.

Cheng, H., & Bouman, C. A. (2001). Multiscale Bayesian segmentation using a trainable context model. *IEEE Trans. On Image Processing,* 10(4):511-525.

Della Pietra, S., Della Pietra, V., & Lafferty, J. (1997). Inducing features of random fields. *IEEE transactions on Pattern Analysis and Machine Intelligence,* 19, 380-393.

Hammersley, J., & Clifford, P. (1971). Markov fields on finite graphs and lattices. Unpublished manuscript.

Kumar, S., & Hebert, M. (2003). Discriminative random fields: A discriminative framework for contextual interaction in classification. *IEEE Int, Conf. on Computer Vision*, 2:1150-1157.

Kumar, S., & Hebert, M. (2004). Discriminative Fields for Modeling Spatial Dependencies in Natural Images. *Advances in Neural Information Processing Systems, NIPS 16*.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proc. ICML*.

Leek, T. (1997). Information extraction using hidden Markov models. Master's thesis, University of California, San Diego.

Li, J., Najmi, A., & Gray, R. M. (2000). Image Classification by a Two-dimensional Hidden Markov Model. *IEEE Trans on Signal Processing,* Vol. 48, No. 2.

Li, S. Z. (2001). Markov Random Field Modeling in Image Analysis, Springer-Verlag, Tokyo.

Liu, B., Grossman, R. & Zhai, Y. (2003). Mining data records in web pages. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Liu, D. C., & Nocedal, J. (1989). ON THE LIMITED MEMORY BFGS METHOD FOR LARGE SCALE OPTIMIZATION. *Mathmetical Programming* 45, pp. 503-528.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. *In Sixth Conf. on Natural Language Learning,* pages 49-55.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proc. ICML 2000,* pp. 591-598.

Murphy, K. P., Weiss, Y., & Jordan, M. I. (2002). Loopy belief propagation for approximate inference: An empirical study. *Fifteenth Conference On Uncertanty in Artificial Intelligence (UAI)* (pp. 467-475).

Nie, Z., Zhang, Y., Wen, J. R., & Ma, W. Y. (2005). Object-Level Ranking: Bringing Order to Web Objects. *To appear in WWW2005*.

Peng, F., & McCallum, A. (2004). Accurate Information Extraction from Research Papers using Conditional Random Fields. *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics*.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE,* 77(2):257–285.

Sha, F., & Pereira, F. (2003). Shallow Parsing with Conditional Random Fields. *Proceedings of Human Language Technology, NAACL 2003*.

Sutton, C., Rohanimanesh, K., & McCallum, A. (2004). Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *ICML*.

Wilson, R., & Li, C. T. (2003). A class of discrete multiresolution random fields and its application to image segmentation. *IEEE Trans. on Pattern Anal. And Machine Intelli.* 25(1):42-56.